electronics

Article

Anti-Jamming Low-Latency Channel Hopping Protocol for Cognitive Radio Networks

Yu-Heng Hsieh, Chih-Min Chao *, Chih-Yu Lin and Chun-Chao Yeh

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung City 202301, Taiwan
* Correspondence: cmchao@ntou.edu.tw

Abstract: In order to increase channel usage efficiency, unlicensed users within a Cognitive Radio Network (CRN) are permitted to utilize channels that are not currently occupied by licensed users. However, ensuring communication between users in a CRN remains a challenge. To overcome this issue, a variety of channel hopping protocols have been developed. Time-invariant channel hopping protocols are vulnerable to attacks, so several channel hopping protocols that are resistant to jamming attacks have been proposed. In the majority of existing anti-jamming protocols, users create their channel hopping sequence using a channel hopping matrix, with the rendezvous probability between two users being determined by the structure of their respective channel hopping matrices. The channel hopping matrices designed by existing methods still have room for improvement. To overcome the difficulty of guaranteeing communication between any pair of users, while also providing protection against jamming attacks and minimizing the time to rendezvous (TTR) in a CRN, this paper presents the Anti-jamming Low-Latency channel hopping (ALL) protocol. This protocol allows a sender to adjust their channel hopping matrix structure to match that of the receiver, thereby improving the chances of successful rendezvous between users. Based on the simulation results, the ALL protocol performs better than the recently proposed practical solution, OLAA, by up to 33% in network throughput and 30% in TTR. On average, ALL outperforms OLAA by 25% in network throughput and 20% in TTR.

Keywords: cognitive radio networks; jamming resistant channel hopping; low latency channel hopping; extended Langford pairing; rendezvous guarantee

check for updates

Citation: Hsieh, Y.-H.; Chao, C.-M.; Lin, C.-Y.; Yeh, C.-C. Anti-Jamming Low-Latency Channel Hopping Protocol for Cognitive Radio Networks. *Electronics* **2023**, *12*, 1811. https://doi.org/10.3390/electronics12081811

Academic Editor: Paul Mitchell

Received: 22 February 2023
Revised: 26 March 2023
Accepted: 31 March 2023
Published: 11 April 2023

1. Introduction

With the rapid development of mobile applications, the traffic load of the unlicensed spectrum has become excessively congested while the licensed spectrum usage remains low [1,2]. To enhance the efficiency of the licensed spectrum, a technology named Dynamic Spectrum Access (DSA), which enables wireless devices to access the licensed spectrum, has been proposed. The DSA technology has been utilized in Cognitive Radio Networks (CRNs). In a CRN, a Primary User (PU) has the priority for channel usage, while a Secondary User (SU) uses a channel that is not occupied by any PU. In a CRN, the rendezvous problem refers to how to enable two SUs to simultaneously use the same channel. There are two types of solutions to the rendezvous problem: using a Common Control Channel (CCC) [3–7] or using a Channel Hopping mechanism [8–21]. For the protocols using CCC, every SU coordinates the channel for data transmission in a specific CCC. Because using a CCC has the issue of a single point of failure and difficulty in finding an available CCC for all SUs due to dynamically changed channel conditions, designing channel hopping protocols is the mainstream solution to the rendezvous problem. An SU running a channel hopping protocol generates a channel hopping sequence to determine the order of using channels. A commonly used criterion for measuring the channel hopping protocols is the time interval between two consecutive rendezvous between two SUs (Time To Rendezvous, TTR). A low

TTR indicates a high rendezvous frequency. In addition to having a low TTR, a well-designed channel hopping protocol should ensure a rendezvous on each of the available channels between any two SUs. The rules for channel hopping for most existing channel hopping protocols are predetermined and do not change during network operation. Such a channel hopping strategy makes the next channel used by an SU predictable and is vulnerable to malicious jamming attacks. Existing rendezvous problem solutions can be divided based on the following two criteria:

- If pre-shared secrets are necessary: Sharing secrets (such as channel hopping sequence) in advance is necessary or not.
- If role pre-assignment is necessary: Assigning a particular role (receiver or sender) to an SU in advance is necessary or not. Symmetric schemes require role pre-assignment, while asymmetric schemes do not.

Based on the above two factors, the classification of existing anti-jamming solutions can be found in Table 1.

Table 1. Existing Anti-jamming Solution Classification.

	with Pre-Shared Secrets	**without Pre-Shared Secrets**
Asymmetric	-	[22–24]
Symmetric	[25,26]	[19–21,27–30], ours

The protocols using pre-shared secrets are suitable for CRNs with fixed topology. However, because the connections between SUs are unstable, the solutions requiring secret sharing in advance are not practical. All of the asymmetric protocols have the problem that SUs of the same role may not have a rendezvous with each other [22–24]. To avoid the issues mentioned above, most anti-jamming protocols are symmetric and do not pre-share secrets.

Many anti-jamming solutions use the channel hopping matrix associated with each SU i to generate its channel hopping sequence [19–21,29,30]. The identifier (ID) of each SU i corresponds to columns in its channel hopping matrix, with each digit of the ID corresponding to a specific column. The transmission and reception modes in the corresponding column are also determined based on the digit. If the IDs of the two SUs are similar, the arrangement of the transmission and reception mode in the two hopping matrixes are also similar. This means that these two nodes stay in the same mode most of the time, resulting in a reduced number of rendezvous. To produce more rendezvous among SUs, we propose the Anti-jamming Low-Latency channel hopping scheme (ALL) to enable a sender to adjust the arrangement of its channel hopping matrix based on its receivers' channel hopping matrixes. The simulation results confirm that, in comparison to the practical solution, OLAA, proposed recently, the suggested method improves network throughput and TTR by as much as 33% and 30%, respectively. On average, ALL achieves better performance than OLAA, with a 25% increase in network throughput and a 20% improvement in TTR.

It should be emphasized that anti-jamming techniques are important and have broad applications in fields involving wireless communication [31], such as military communications, satellite communications, and cyber-physical systems. For example, in cyber-physical power systems, where physical and digital components are integrated to optimize power generation and distribution, dependable and secure communication is vital [32]. Hence, anti-jamming approaches could be relevant in this field to ensure communication channel integrity and availability.

The rest of this paper is organized as follows. In Section 2, we provide a review of the current solutions for the rendezvous problem. Section 3 introduces the preliminary and system models, while Section 4 describes the proposed solution. In Section 5, we present the simulation results, and in Section 6, we provide the conclusions and discuss future work.

2. Related Work

Pre-shared secrets have been utilized by several researchers to propose anti-jamming channel hopping protocols [25,26]. The circular dependency issue may arise in these protocols, as the exchange of pre-shared secrets between SUs may be vulnerable to jamming attacks [30]. The majority of anti-jamming channel hopping protocols do not pre-share secrets and pre-assign roles. To resist jamming attacks, those protocols usually use the available channels randomly. In the UFH protocol, the sender and the receiver choose the channel being used randomly, while a communication link can be built when they simultaneously stay at the identical channel [27]. An SU adjusts the channel hopping sequence based on the strategy of the attacker in the MDP protocol. When the attack strategy is unknown, SUs utilize maximum likelihood estimation and Q-learning to infer the attack strategy [28]. Both the UFH and MDP protocols are able to resist jamming attacks. However, it cannot be ensured that all pairs of SUs will meet within a specific time period.

A rendezvous between any two SUs within a time period is guaranteed in the Sec-CH protocol [29]. SUs running Sec-CH can operate in either transceiving-mode random jump pattern (T mode) or reception-mode stay pattern (R mode). When operating in T mode, an SU is capable of both transmitting and receiving packets, whereas, in R mode, only receiving is allowed. The generation of the channel hopping sequence for an SU a relies on the use of its corresponding channel hopping matrix, denoted as M_a. M_a is constructed such that each of its columns alternates between T-sub-columns (when SU a is in T-mode) and R-sub-columns (when SU a is in R-mode). The two distinct types of columns in M_a are bit-0 column (which begins with a T-sub-column) and bit-1 column (which begins with an R-sub-column). Each SU is assigned a unique binary ID, and the value of each digit within the ID of SU a corresponds to a specific column within M_a. Specifically, if the value of the digit is 0, a bit-0 column is selected, and if the value of the digit is 1, a bit-1 column is chosen. In a load-balanced environment, Sec-CH has demonstrated high performance. The Tri-CH protocol, an improved version of the Sec-CH protocol, reduces the Maximum Time To Rendezvous (MTTR) of Sec-CH [30]. Every SU that operates using Tri-CH constructs its unique hopping matrix by utilizing its respective ternary ID. Both the J mode and S mode are utilized to create each sub-column within the hopping matrix. Each digit in the ID of an SU corresponds to one of three different sub-columns, with possible values of 0, 1, and 2. Like Sec-CH, Tri-CH may not perform well in a network where the traffic load is imbalanced.

In networks with unbalanced traffic loads, the LAA protocol has demonstrated good performance [19]. An SU running LAA constructs its hopping matrix based on the extended Langford pairing (ELP) [33]. When an ELP sequence has an order of n_e, an SU that utilizes the LAA protocol is given an (n_e + 1)-ary ID. The construction of a channel hopping matrix M_a involves forming each of its columns by combining a transmission part denoted as the T frame, and a reception part denoted as the R frame. The ratio and arrangement of these two parts are determined by the traffic load of SU a as well as the specific digit in its (n_e + 1)-ary ID. Both the OLAA_T and OLAA_R protocols are improved versions of the LAA protocol. Each SU running OLAA_T and OLAA_R adjusts its usage probability for each channel based on the probability of PU occupancy on that channel, with OLAA_T focusing on the transmission part and OLAA_R focusing on the reception part. Such a strategy makes OLAA_T and OLAA_R practical anti-jamming solutions [21]. Specifically, the ratio of the T frame to the R frame is adjusted based on their own traffic load.

$$p_{a,i} = \frac{(1 - O_{a,i})}{\sum_{j=1}^{N_c}(1 - O_{a,j})} \tag{1}$$

An SU constructs its channel hopping matrix based on its ID. The channel hopping matrix of an SU is constructed with $L + 1$ columns, where each column comprises multiple sub-columns, given an SU ID length of L. In OLAA_T, each column of the channel hopping matrix M_a is composed of k_a sub-columns, and each sub-column contains $2(n_e + 1)$ frames, with each frame containing $4N_c$ slots. The value of k_a is equal to the number of available channels for SU a. Each T frame includes $2N_c$ default slots and $2N_c$ adjustment slots. The $2N_c$ default slots are divided into two groups, each of which has N_c consecutive slots. A total of N_c channels are assigned to each group. If a channel is unavailable, a randomly selected available channel is substituted. During each adjustment slot, a channel is chosen based on its usage probability, which is determined using Equation (1). In each R frame, all slots are assigned the same channel. In the same sub-column, the R frames are assigned the same channel. However, the R frames in different sub-columns are assigned different channels, requiring k_a sub-columns to allocate all available channels for SU a.

In OLAA_R, there are k_a default sub-columns and k_a adjustment sub-columns in each column of the channel hopping matrix M_a. Each sub-column contains $2(n_e + 1)$ frames, and each frame has $2N_c$ slots. In each T frame, $2N_c$ default slots are divided into two groups, with each group having N_c consecutive slots. The channel assignment in the T frames, as well as the R frames in the default sub-column, remain the same as in OLAA_T. However, for the R frames in the adjustment sub-columns, SU a selects channel i with a probability obtained using Equation (1).

3. Preliminary and System Model

We proposed a symmetric and without pre-shared secrets solution in this paper. In the considered CRN, there are N_c channels with the channel set C, $C = \{1, 2, 3, \ldots, N_c\}$. The SUs are randomly deployed across the network, and their traffic loads vary. SU a is identified by an $(n_e + 1)$-ary ID and k_a available channels. Let C_a be the available channel set of SU a. There is at least one commonly available channel between any two SUs. Three kinds of jammers are randomly deployed in the CRN [17,18,20,25,27]:

- Static jammers: They will consistently jam a fixed channel.
- Arbitrary jammers: They will jam a channel chosen randomly.
- Clever jammers: Each intelligent jammer will have a cognitive radio, and it will jam the channel most frequently used by a particular SU.

The key variables used in this paper are presented in Table 2.

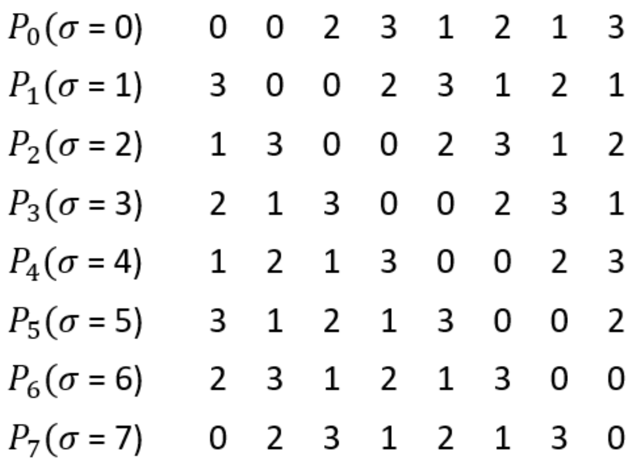
Table 2. Variable listing.

Variable	Definition
n_e	order of an ELP
N_u	number of SUs
N_c	number of channels
C	channel set
k_i	number of available channels of SU i
C_i	available channel set of SU i
ID_i	SU i's $(n_e + 1)$-ary ID
ID_i^n	nth digit of SU i's $(n_e + 1)$-ary ID
R_i	receiver set of SU i
R_i^n	nth receivers of SU i
M_i	SU i's hopping matrix
M_i^n	nth column of M_i
FR_i^n	fixed R frame symbol of M_i^n
uFR_i	unfixed R frame symbol set in each column of M_i
uFR_i^n	unfixed R frame symbol set of M_i^n
$uFR_i^{n,x}$	xth unfixed R frame symbol of the nth column of M_i
e_i^n	ELP pattern of M_i^n
$e_i^{n,x}$	xth symbol in the ELP pattern of M_i^n
CHS_i	SU i's channel hopping sequence
NT_i	number of T frames of SU i
NR_i	number of R frames of SU i
L_c	length of a column in a channel hopping matrix
λ_i^m	ID digit corresponding to M_i^m

In the considered CRN, time is divided into several equal-sized slots. An SU switches channels based on its channel hopping sequence. The proposed protocol uses the ELP sequence to generate the channel hopping sequence of each SU.

- Extended Langford pairing

A Langford pairing (LP) is a permutation of the sequence of $2n$ symbols, where n is the order of the Langford pairing [34,35]. In a LP of order n, there are n distinct symbols, each of which appears twice. The distance between two identical symbols is equal to the symbol's value. An example LP of order 3 is $(2,3,1,2,1,3)$. Appending two zeros at the beginning of an LP produces an extended LP (ELP) [33]. The ELP of order 3 built from the above LP is $(0,0,2,3,1,2,1,3)$. Let n_e be the order of an ELP sequence. The rotated ELP sequences are considered to be different patterns. The different patterns produced by rotating the ELP sequence $(0,0,2,3,1,2,1,3)$ to the right by σ times with $n_e = 3$ is shown in Figure 1.



$P_0(\sigma = 0)$	0	0	2	3	1	2	1	3
$P_1(\sigma = 1)$	3	0	0	2	3	1	2	1
$P_2(\sigma = 2)$	1	3	0	0	2	3	1	2
$P_3(\sigma = 3)$	2	1	3	0	0	2	3	1
$P_4(\sigma = 4)$	1	2	1	3	0	0	2	3
$P_5(\sigma = 5)$	3	1	2	1	3	0	0	2
$P_6(\sigma = 6)$	2	3	1	2	1	3	0	0
$P_7(\sigma = 7)$	0	2	3	1	2	1	3	0

Figure 1. Different ELP patterns.

4. Proposed Solution

An SU employing the ALL protocol generates a channel hopping matrix by utilizing its ID and the chosen ELP pattern. Let M_a represent the channel hopping matrix of user a, and M_a^n denote the nth column of M_a. The duration required to complete the entire channel hopping sequence constructed from the respective channel hopping matrix is referred to as a cycle. Using the ALL protocol, each user divides time into continuous periods, each consisting of several cycles. The sequence of channel hopping denoted as CHS_a is generated by SU a by selecting the elements of its associated channel hopping matrix M_a sequentially in a row-wise manner. Each column of M_a consists of multiple frames of T and R, and each frame is composed of several time slots. During time slots assigned to T frames, an SU is allowed to transmit or receive packets, while during time slots assigned to R frames, it can only receive packets. In order to improve the rendezvous probability between a sender and its intended receiver, an SU running ALL has the capability to adjust the order of T and R frames in its channel hopping matrix to match that of the receiver.

4.1. The Proposed ALL Protocol

In most existing channel hopping algorithms, the ID of each SU is set to its 48-bit MAC address. The initial 24 bits of a MAC address denote the manufacturer or vendor number, whereas the remaining 24 bits signify a distinctive serial number assigned to the specific network interface controller by the manufacturer. To increase the dissimilarity of IDs among different SUs, an SU a set the last few bits of the MAC address as its ID (denoted as ID_a). Recall that an SU uses $(n_e + 1)$-ary ID in the ALL protocol. To obtain the appropriate length for ID in a distributed manner, each SU uses the existing 3B algorithm [36] to find the number of nodes N_u in the network and then broadcasts N_u to synchronize with other SUs. The ID length of SU a can be obtained by Equation (2).

$$|ID_a| = \lfloor log_{n_e+1} N_u \rfloor \tag{2}$$

The channel hopping matrix M_a is composed of $|ID_a| + 1$ columns. The first column is composed of a repetitive pattern of frames, specifically $TTRR$, wherein T denotes a T frame, and R denotes an R frame. Each of the other $|ID_a|$ columns consists of k_a sub-columns. The sequence of T and R frames in every column is established using an ELP pattern that is chosen according to the corresponding digit of ID_a. Let $e_a^n = ID_a^n, 1 \leq n \leq |ID_a| + 1$ be the ELP pattern of the nth column of M_a. Every sub-column is comprised of $2(n_e + 1)$ frames. A frame in each sub-column sequentially corresponds to a symbol in the ELP pattern. Through the use of the OLAA_T protocol, SU a establishes the quantity of T and R frames (represented by NT_a and NR_a, respectively) in each sub-column according to its traffic load. In each T frame, $2N_c$ default slots and $2N_c$ adjustment slots are arranged alternately. The channel assignment is also achieved by applying OLAA_T. Specifically, N_c channels will be randomly and non-repeatedly assigned to the first-half of default slots as well as the second-half ones (channels that are unavailable are substituted with available channels in a random manner). The usage probability of each channel is calculated according to Equation (1). The channel allocation in the adjustment slots is based on the usage probability. Consisting of $4N_c$ slots, each R frame within a given sub-column is assigned to the same channel. Distinct channels are assigned to R frames in separate sub-columns of a single column.

To demonstrate the channel hopping matrix generation and channel allocation for each frame, consider the example of SU a ($ID_a = 00_4$) utilizing the ELP sequence (0, 0, 3, 1, 2, 1, 3, 2) to create a channel hopping matrix with $C = C_a = 1, 2$. It is assumed that channel 1 has a PU occupancy of 80% and channel 2 has a PU occupancy of 40%. Additionally, it is assumed that SU a has a balanced amount of incoming and outgoing traffic. Because $ID_a = (00)_4$, $e_a^0 = e_a^1 = (0, 0, 3, 1, 2, 1, 3, 2)$ while $NT_a = NR_a$ since SU a has balanced traffic, as shown in Figure 2a. The channel assignment of frames 4 and 6 of sub-column 0 and frame 7 of sub-column 1 in M_a^3 can be found in Figure 2b. In frame 4 of sub-column 0, the sequence generated by a random arrangement of elements in C by SU a are (1, 2) and (2, 1), respectively, and thus the default slots of the first- and second-half of the frame will be assigned channels (1, 2) and (2, 1), respectively. SU a assigns channels in C to the R frame of each sub-column randomly and without duplication. In this example, SU a assigns channel 1 and channel 2 to sub-column 0 and sub-column 1, respectively. The channel assignment to the adjustment slots of each T frame is determined based on Equation (1), which means that the probability that SU a assigns channels 1 and 2 to the adjustment slot is $\frac{(1-O_{a,1})}{((1-O_{a,1})+(1-O_{a,2}))}$ = 25% and $\frac{(1-O_{a,2})}{((1-O_{a,1})+(1-O_{a,2}))}$ = 75%, respectively.

When a sender SU a is in a slot belonging to a T frame and, at the same time, receiver SU b is in a slot belonging to an R frame, we call SUs a and b TR *overlapped*. Because each available channel is being used at least once in a T frame, every $2N_c$ slots and the channel utilized by an R frame remains unchanged, two SUs are guaranteed to have a rendezvous when they are TR overlapped for more than $2N_c$ slots. To facilitate the rendezvous guarantee, each SU a running ALL divides R frames into *fixed R frames* and *unfixed R frames*. An SU first determines the fixed R frame symbol of each column based on its ID, which also determines the position of the fixed R frame in each sub-column. Specifically, each SU a determines the fixed R frame symbol of the corresponding column in M_a according to the value of each digit of ID_a. The fixed R frame symbol of the $(n + 1)$th column in M_a, denoted as FR_a^{n+1}, is calculated as $FR_a^{n+1} = ID_a^n$. When the ELP symbol in M_a^n is equal to FR_a^n, the frame corresponding to the symbol is set as the fixed R frame. For example, suppose that SU a uses the ELP sequence $(0, 0, 3, 1, 2, 1, 3, 2)$ to build M_a, ID_a=$(00)_4$, $e_a^1 = e_a^2 = (0, 0, 3, 1, 2, 1, 3, 2)$. Since $FR_a^2 = FR_a^3 = 0$, the position of the fixed R frame in the second and the third columns of M_a are both at the frames corresponding to symbol 0 in each sub-column, that is, frames 0 and 1.

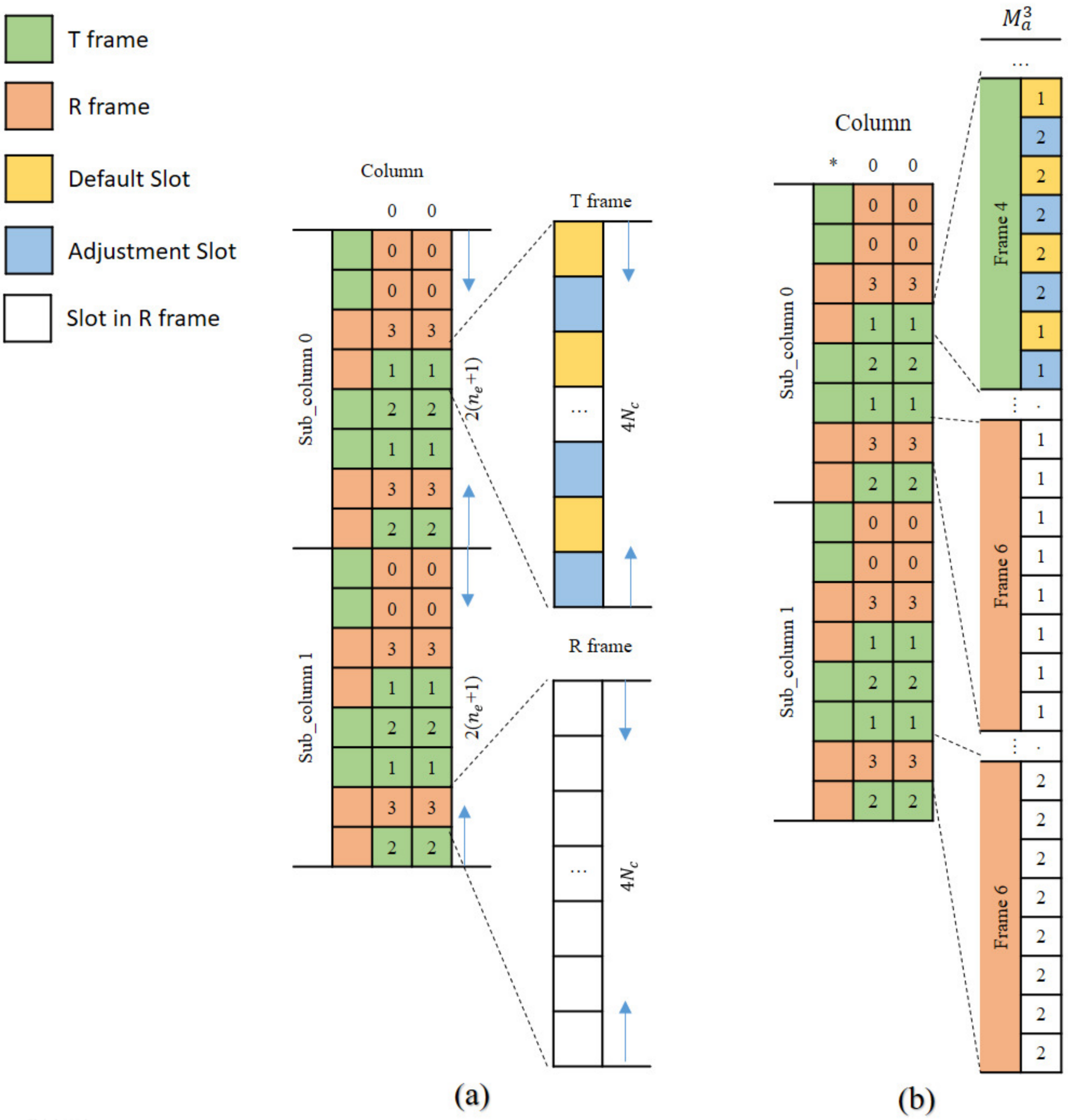


Figure 2. The hopping matrix structure of SU a is presented in two parts: (**a**) the allocation of T and R frames and slots, and (**b**) the channel allocation for frames 4 and 6 of Sub_column 0 and frame 7 of Sub_column 1 in M_a^3.

Each SU a has $NR_a - 1$ unfixed R frame symbols in each sub-column. The set of all the unfixed R frame symbols in M_a are represented by uFR_a. The set of all unfixed R frame symbols in M_a^n is represented by uFR_a^n, while the xth unfixed R frame symbol is represented by $uFR_a^{n,x}$. Different SUs may be out of synchronization, that is, the time at which each SU starts the channel hopping sequence may be different. When a sender allocates its unfixed R symbols, the time difference with the receiver must be considered. Suppose that sender SU a connects to network θ time slots prior to its receiver, SU b. In such a scenario, when SU a switches to the channel located in the αth column of M_a, SU b switches to the βth column of M_b, where $\beta = (\theta + \alpha) \bmod (|ID_a| + 1)$. In each corresponding column of two SUs, the γth frame of SU a overlaps $2N_c$ slots with the δth frame of SU b, where

$$\delta = \begin{cases} (\gamma + \frac{\frac{\theta}{|ID+1|}}{4N_c}) \pmod{2(n_e+1)}, & \text{if } 0 \leq \beta \pmod{4N_c} \leq\ 2N_c \\ (\gamma + \frac{\frac{\theta}{|ID+1|}}{4N_c} + 1) \pmod{2(n_e+1)}, & \text{otherwise.} \end{cases}$$

SU a determines the unfixed R frame symbol of M_a in a column-by-column manner. The unfixed R symbols for the second column of M_a are the $NR_a - 1 \pmod{(n_e+1)}$ symbols next to FR_a^2. The unfixed R symbols for the third column of M_a are the $NR_a - 1 \pmod{(n_e+1)}$ symbols next to the last unfixed R symbol in the previous column of M_a. The unfixed R symbols for the other columns of M_a can be obtained by analogy.

The algorithm of the ALL protocol is shown in Algorithm 1. To guarantee rendezvous with other SUs, a period consists of $NR_a - 1$ channel hopping cycles (line 1). A repetitive pattern of frames, $TTRR$, makes up the first column (line 2). Each SU a takes the last $(\lfloor \log_{n_e+1} N_u \rfloor + 1)$ bits of its MAC address as ID_a (line 3). The unfixed R frame symbol for each column are selected according to the positions of all fixed R frame symbols of the receivers (line 4–10), while the channels are assigned for all T and R frames (line 11). The channel hopping sequence is generated by picking elements of the channel hopping matrix in row-major order (line 12).

Algorithm 1 ALL

```
Input: $n_e, N_u, \theta$
Output: $CHS_a$
1:  for $k \in [1, |NR_a - 1|]$ do
2:      Assign frames by the repeated $TTRR$ pattern for the first column of the hopping
        matrix;
3:      Let the last $(\lfloor \log_{(n_e+1)} N_u \rfloor + 1)$ bits of MAC address be $ID_a$;
4:      for $m \in [1, |ID_a|]$ do
5:          for $n \in [1, |R_a|]$ do
6:              Find $R_a^n$, $\alpha$ and $\delta$ of $SU_a$.
7:          end for;
8:          Assign the $NR_a - 1 \pmod{(n_e+1)}$ symbols next to symbol $w$ as $uFR_a^m$ where
        $w = FR_a^m + k \times (NR_a - 1) \pmod{(n_e + 1}$.
9:          Set the symbols not being assigned to $FR_a^m$ or $uFR_a^m$ as $T$ frame symbols.
10:     end for;
11:     Apply OLAA_T to allocate channels for $T$ and $R$ frames;
12:     Each node hops by row order of the channel hopping matrix;
13: end for;
```

Note that a period for SU a is equal to $(NR_a - 1) \times L_c \times (|ID_a| + 1)$ time slots, where $L_c = k_a \times 4N_c$.

An example is utilized below to illustrate the operation of the ALL protocol. The following assumptions are made in this example: Assume that the sender, SU a, and the receiver, SU b, are load balanced with $C = C_a = C_b = 1, 2$. SUs a and b with $ID_a = (01)_4$ and $ID_b = (22)_4$ use the ELP sequence (0, 0, 3, 1, 2, 1, 3, 2) to build their channel hopping matrixes while $e_a^1 = (0,0,3,1,2,1,3,2)$, $e_a^2 = (2,0,0,3,1,2,1,3)$, $e_b^1 = e_b^2 = (3,2,0,0,3,1,2,1)$. SU a joins the network one slot earlier than the receiver SU b. Since SUs a and b are load balanced, we have $NT_a = NR_a = NT_b = NR_b = 2$. Note that $FR_b^1 = FR_b^2 = \{2\}$, which means that the fixed R frames of the second and third columns of M_b are frames 1 and 6, as shown in Figure 3a. When $\theta = 1$, the M_a^3 and M_b^2 have a rendezvous and thus SU a try not to assign an unfixed R frame in M_a^3 in frames 1 and 6 in each sub-column. Assume that SU a assigns frames 0 and 5 as unfixed R frames, as shown in Figure 3b. Because M_a^2 and M_b^1 have a rendezvous and M_b^1 is built from the repeated (TTRR) pattern, SU a will assign the consecutive $NR_a - 1(= 1)$ symbol next to $FR_a^2 (= \{0\})$ as $uFR_a^2 = \{1\}$, as shown in Figure 3c. The complete M_a is shown in Figure 3d.

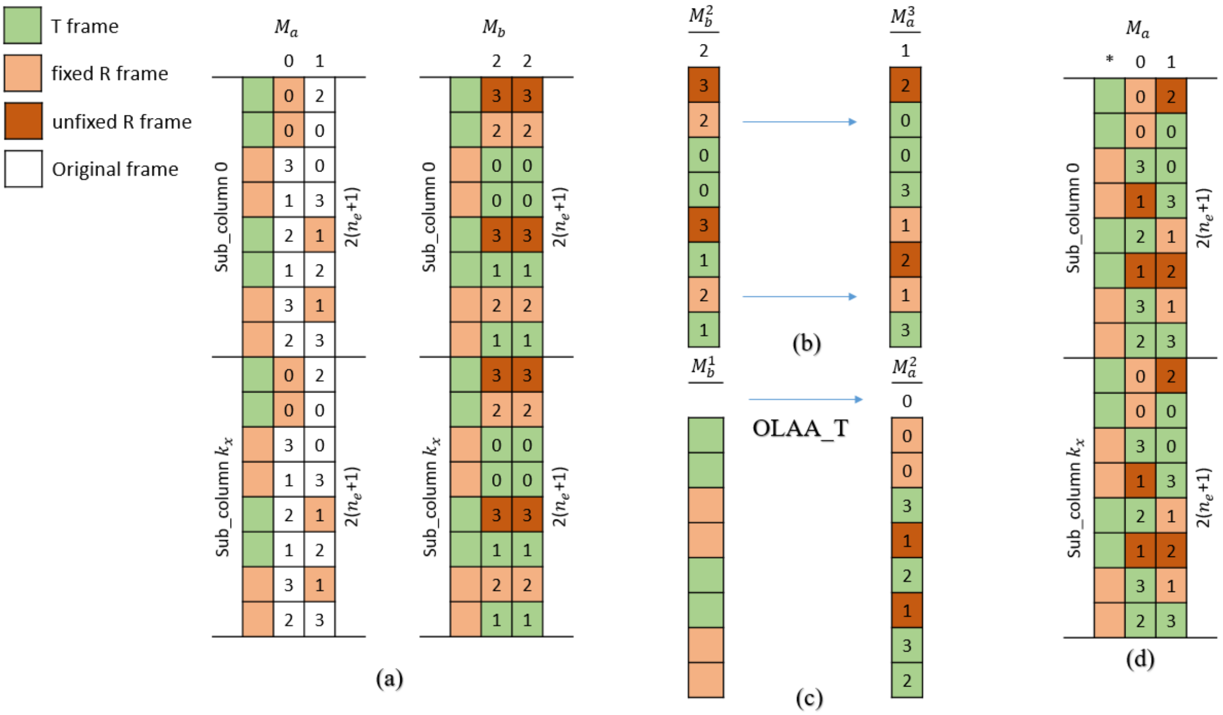


Figure 3. This figure illustrates how SU a allocates T and unfixed R frame symbols in the presence of an asynchronous SU b. The four panels depict the following steps: (**a**) the allocation of uFR_a, (**b**) the selection of T frame and unfixed R frame symbol of M_a^3 based on M_b^2, (**c**) the selection of T frame and unfixed R frame symbol of M_a^2 based on M_b^1, and (**d**) the complete M_a.

4.2. Property of the ALL Protocol

SUs using the ALL protocol are guaranteed to have a rendezvous with their intended receiver, as described in Theorem 1.

Theorem 1. *The ALL protocol ensures that an SU a and its designated receiver b have a rendezvous on all mutually available channels during a given period.*

Proof. The channel hopping sequence of SU a is produced by the channel hopping matrix M_a, the components in the same column of M_a will appear in the channel hopping sequence every $(|ID_a| + 1)$ slots. We can prove that the channel hopping sequence of SU a is guaranteed to have a rendezvous with SU b by proving that there is TR overlapping between SUs a and b. With the loss of generality, we assume the ith bit of ID_a and that of ID_b are different. When SUs a and b are synchronous, SU a uses channels in M_a^i while SU b uses channels in M_b^i. Because $ID_a^i \neq ID_b^i$, the ELP pattern and thus the fixed R frame symbol selected by SU a must be different from that selected by SU b. It means that, at most one, fixed R frame of SU a overlaps with that of SU b. For each column other than the first one in a hopping matrix, SU a will assign each ELP symbol other than its fixed R symbol in the column as a T frame in a period. This means that SU a can assign a T frame to overlap with a fixed R frame of SU b. Whenever SUs a and b are TR overlapped, they will have a rendezvous on each of their available channels since SU b uses different channels for fixed R frames in different sub-columns.

There are two cases when SUs a and b are asynchronous:

Case 1: $\theta \neq 0 \pmod{(n_e + 1)}$. In this case, without loss of generality, assuming that SU a uses the channels in M_a^j where $j \neq 1$ when SU b uses channels in M_b^1. Because M_b^1 is

generated by the repeated frame pattern (TTRR), which is different from the T and R frame arrangement in M_a^j, SUs a and b are TR overlapped.
Case 2: $\theta = 0$ (mod $(n_e + 1)$). The proof, in this case, is the same as the one when SUs a and b are synchronous.

We have demonstrated that the theorem is valid, which states that SU a and its designated receiver b will have a rendezvous on all mutually available channels within a given period, regardless of whether they are synchronous or asynchronous. □

5. Simulation

We have utilized NS-3 (version 3.17) to execute the ALL, OLAA_R, and OLAA_T protocols and validate their effectiveness. The simulation is conducted in a square environment with a length of 1 km for each side. A two-state Markov chain is used to model the PU activity of each channel. Each PU switches between idle and busy states where the interval between two consecutive activation points and the interval of idle time are both exponentially distributed with a mean of two time slots. There are also 8 PUs, 30 SUs, and 21 malicious attackers in the network. The number of channels being used in the network equals the number of PUs. The MAC protocol used in the simulation is IEEE 802.11 CSMA/CA. The transmission range of SUs and attackers is set to 250 m. Table 3 presents the essential simulation parameters used in the experiments. Figure 4 depicts the three different topologies, namely mesh, star, and bottleneck, which are utilized to simulate diverse traffic loads. For the mesh topology, the SUs are distributed randomly in the network, and each SU randomly selects one of its neighbors as the receiver. In the star topology, a central node is located at the center of the network, and all other nodes are randomly distributed in the network and send their packets to the central node. In the bottleneck topology, certain nodes are designated as bottleneck nodes with more incoming traffic than outgoing traffic, such as nodes A and B, shown in Figure 4c. Other nodes in the network transmit their packets to the bottleneck node they are connected with on the right. The network was also populated with three types of malicious attackers that were randomly distributed, with each type accounting for one-third of the total number of attackers. Each point in the following figures is the average of 100 simulation runs. In the simulations, a common ELP sequence of order three (0, 0, 3, 1, 2, 1, 3, 2) is used by each SU to construct the channel hopping matrix for every node that employs the ALL and OLAA_T protocols. This implies that, for SU a running ALL, the maximum NR_a is equal to three. In the simulations, the length of ID for SUs running OLAA is set to 24 bits. The simulation time is set to $(24 + 1) \times k_a \times 2 \times (3 + 1) \times 4N_c = 800 \times (k_a \times N_c)$ time slots, which is the time span of a cycle in the OLAA_T protocol and 6.25 cycles in the ALL protocol. As rendezvous is guaranteed in $NR_a - 1$ cycles, such a simulation time is long enough for SUs running ALL to have a rendezvous with their receivers. The metrics we observed include the accumulated throughput of all the SUs in the network during the simulation time and the average time to rendezvousing (ATTR) between all transmission pairs.

Table 3. Simulation Parameters.

Parameter	Value
Area size	1000 × 1000 m
Number of PUs	8
PUs Idle ratio	50%
Number of SUs	30
Transmission range of SU/jammers	250 m
Packet size	512 bytes
Simulation runs	100
Simulation times	$800 \times k_a \times N_c$ slots

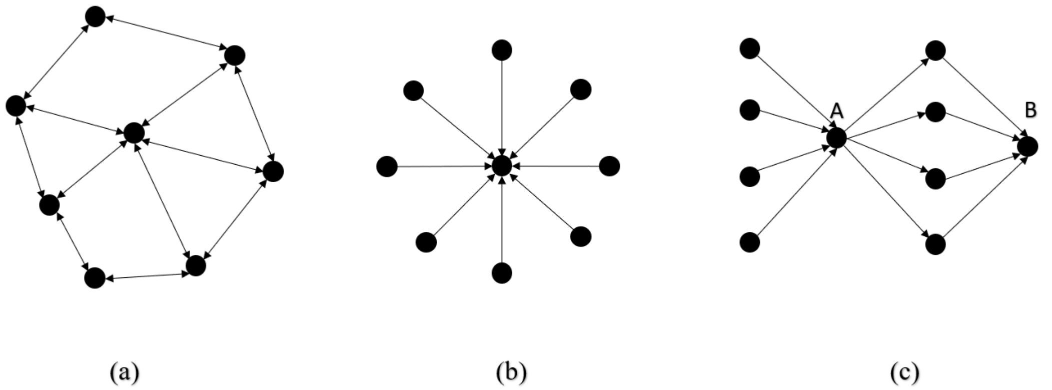


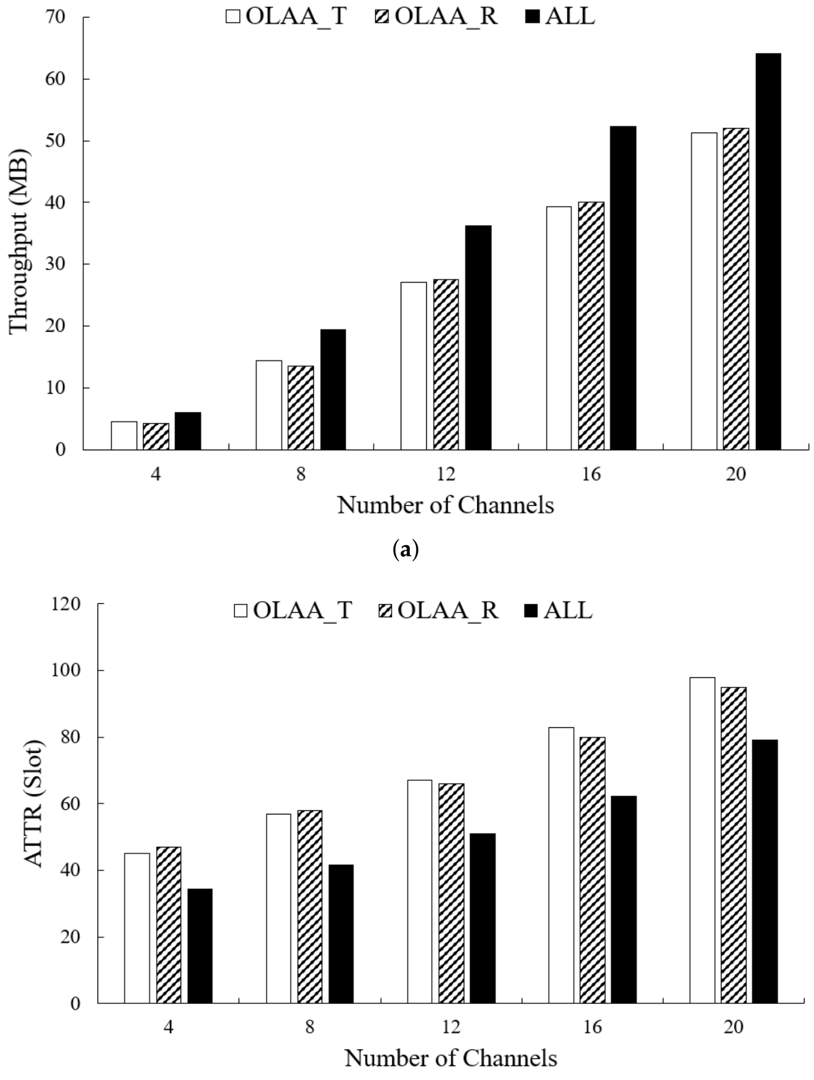
Figure 4. This figure illustrates the three network topologies employed in the simulations: (**a**) mesh, (**b**) star, and (**c**) bottleneck.

The performance of the proposed protocol is observed from the following three aspects.

(A) Impact of the number of channels: In this experiment, we change the number of channels in the network. For each SU running ALL, the size of M_a is changed as varied k_a, and the simulation time is changed accordingly to provide a guaranteed rendezvous. The 21 malicious attackers are randomly distributed in the network in this experiment. The simulation results of mesh, bottleneck, and star topologies for synchronous and asynchronous networks are shown in Figures 5–7 and Figures 8–10, respectively. As the number of channels increases, the chance of SUs having a rendezvous is reduced, and thus, the ATTR values in all three topologies gradually decrease. The ALL protocol performs better than OLAA_T and OLAA_R in both synchronous and asynchronous networks because each SU running ALL can adjust the positions of T frames according to the fixed R frames of its intended receivers to increase the number of guaranteed rendezvous. Specifically, when compared to the OLAA_T and OLAA_R protocols, the ALL protocol achieves an average of 25% and 20% improvement in terms of throughput and TTR, respectively.

The simulation results also reveal that the ALL protocol has similar performance in synchronous and asynchronous networks. When SUs are synchronous, the sender and its receiver have a rendezvous only in the columns where their corresponding ID digits are different. That is, two nodes have many rendezvous when their IDs are very different. When SUs, say sender a and receiver b, are asynchronous but the first column of M_a overlays with that of M_b, the rendezvous condition is similar to that of a synchronous network. When the first column of M_a does not overlay with that of M_b, there is a guaranteed rendezvous between the first column of M_a and a column other than the first one in M_b. SUs a and b will have many rendezvous when their IDs corresponding to other columns are very different. Because the node IDs are uniformly distributed in the simulations, similar numbers of rendezvous can be found for synchronous and asynchronous networks.

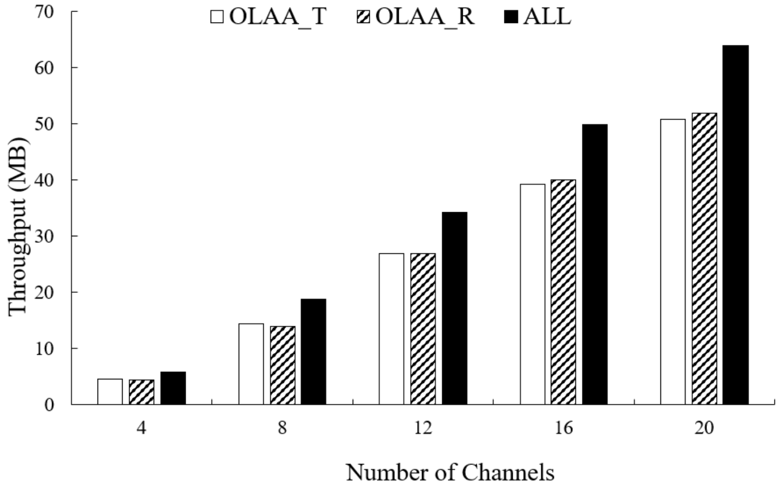
The ALL protocol also achieves similar performance in different topologies. Note that the ALL protocol outperforms the OLAA_T and OLAA_R protocols in all three different topologies. The central node of the star topology and the bottleneck node in the bottleneck topology suffer from transmission collisions, and thus, the produced ATTR values are higher than that of the mesh topology. For example, when 20 channels are used in a synchronous network, the throughput for the mesh, bottleneck, and star topologies is 64.1, 63.8, and 63.9 MB, respectively. The ATTR in the same environment for the three topologies is 78.5, 79.1, and 79.0 slots, respectively. When 20 channels are used in an asynchronous network, the throughput for the mesh, bottleneck, and star topologies is 64.1, 63.7, and 63.9 MB, respectively, while the ATTR for three topologies is 78.5, 79.1, and 79.0 slots, respectively. Since the performance in different topologies is similar and whether SUs are synchronous has little impact on performance, to avoid redundancy, we present the results of the mesh topology with synchronized SU nodes in the following simulations.



(**a**)

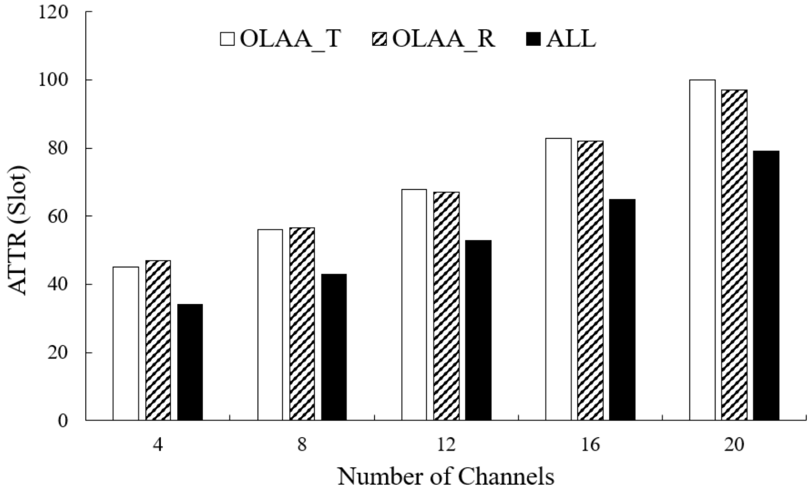
(**b**)

Figure 5. This figure displays the impact of the number of channels on the mesh topology's performance in a synchronous network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.



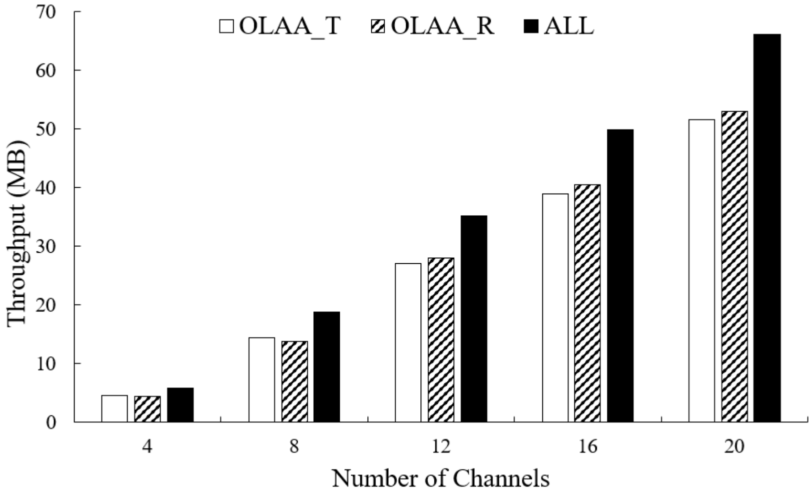
(**a**)

Figure 6. *Cont.*

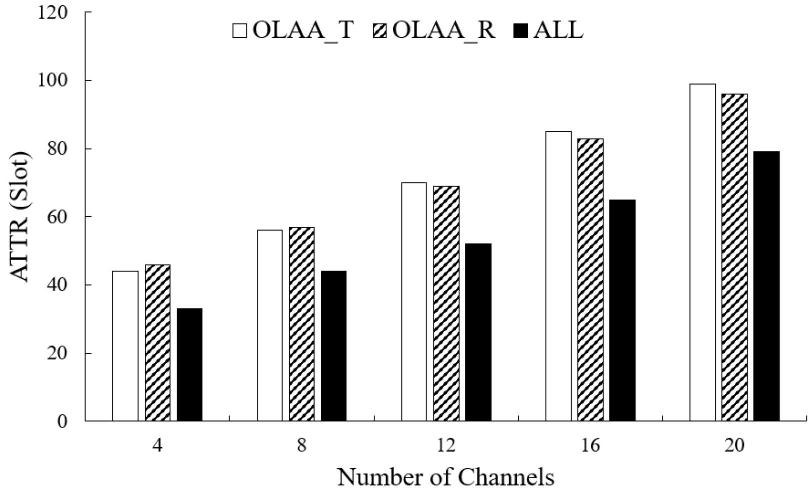


(**b**)

Figure 6. This figure displays the impact of the number of channels on the bottleneck topology's performance in a synchronous network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.

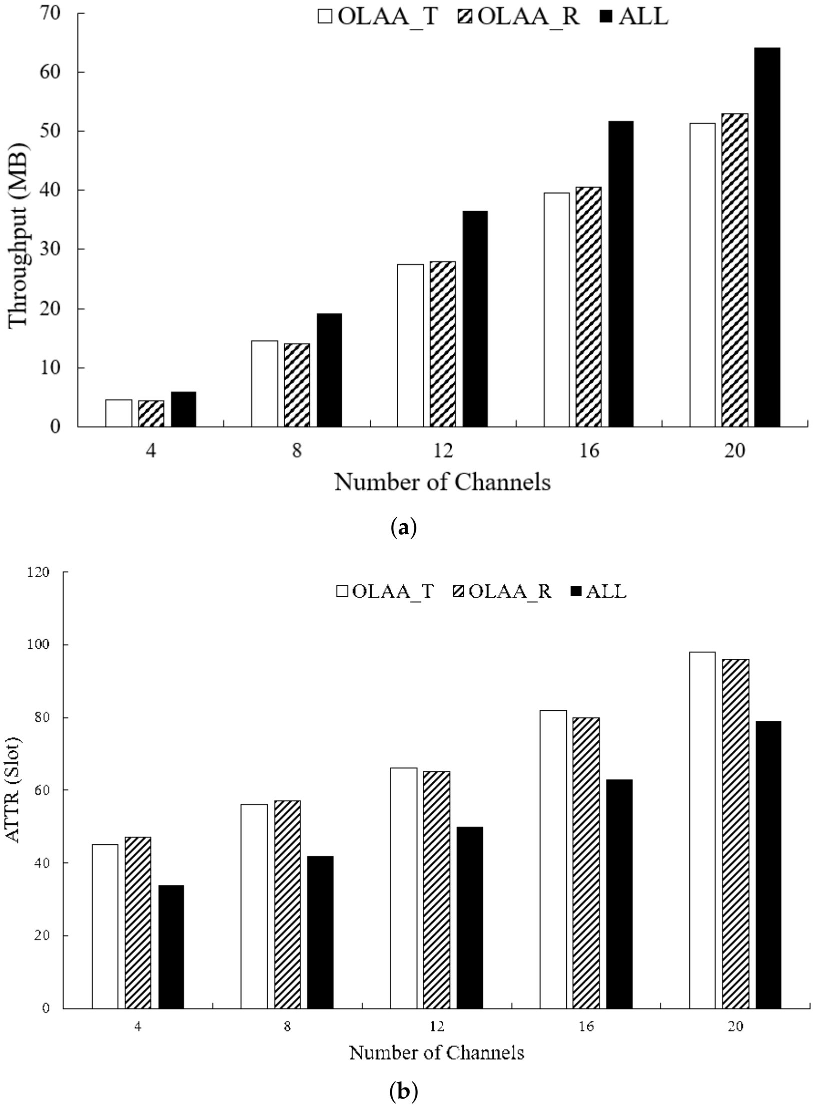


(**a**)



(**b**)

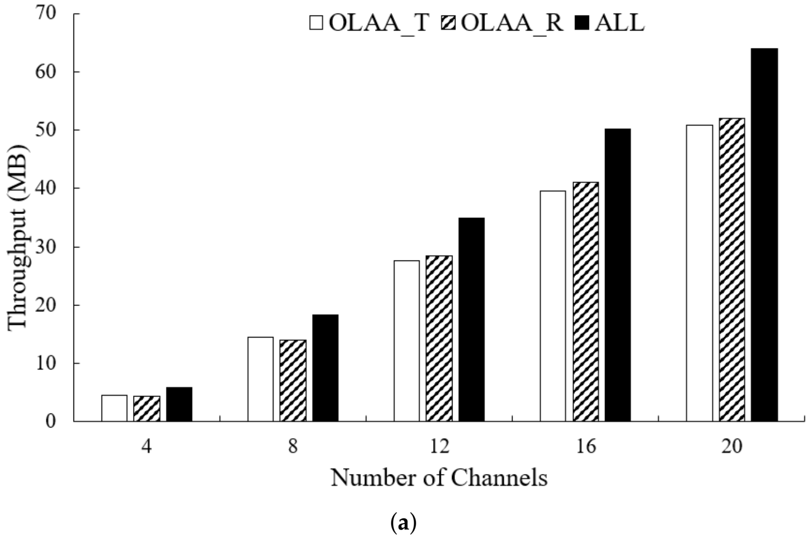
Figure 7. This figure displays the impact of the number of channels on the star topology's performance in a synchronous network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.



(**a**)

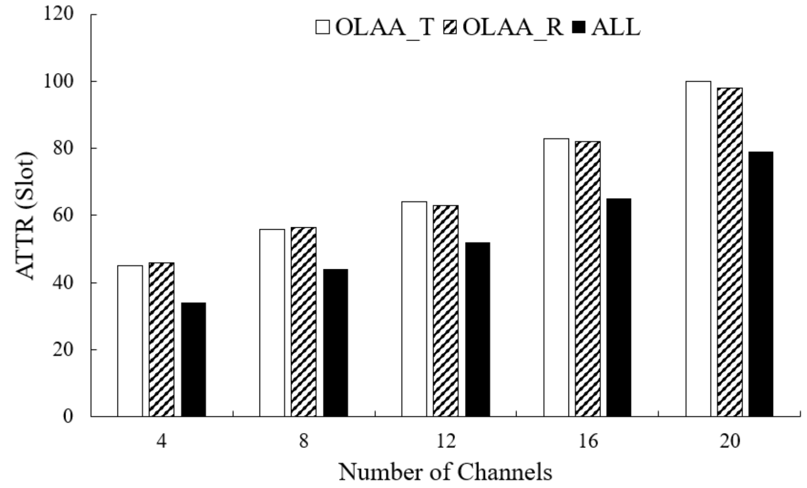
(**b**)

Figure 8. This figure displays the impact of the number of channels on the mesh topology's performance in an asynchronous network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.



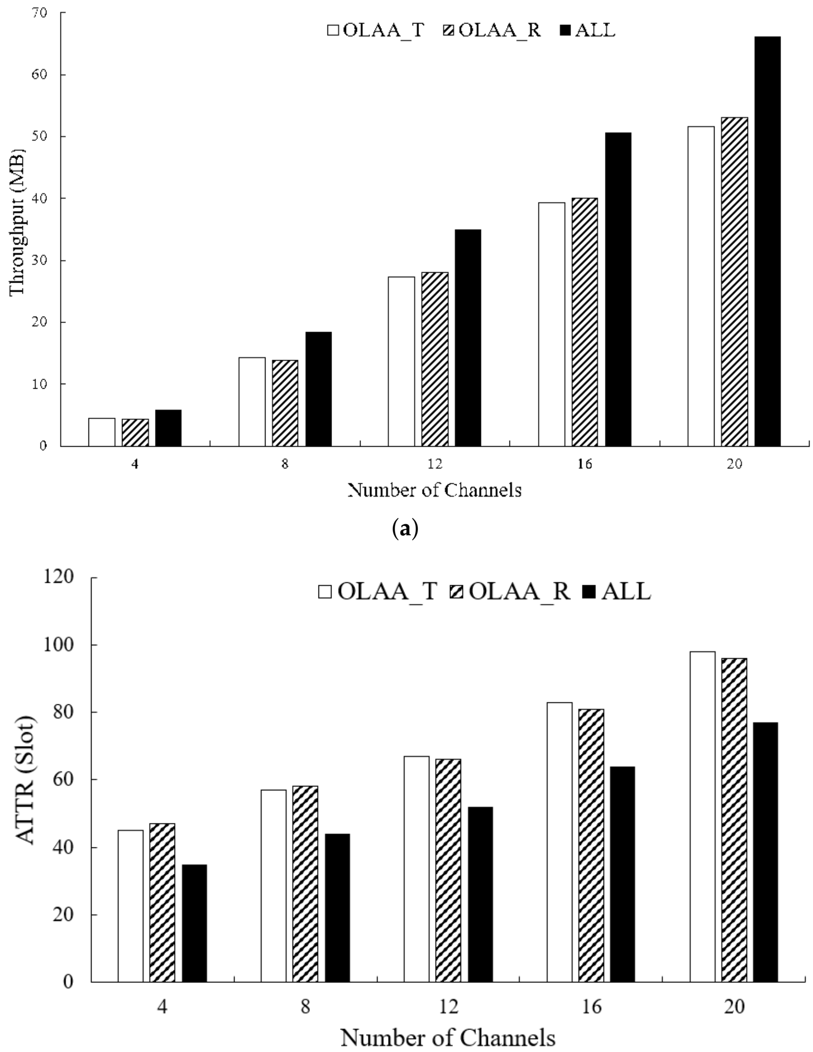
(**a**)

Figure 9. *Cont.*



(**b**)

Figure 9. This figure displays the impact of the number of channels on the bottleneck topology's performance in an asynchronous network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.

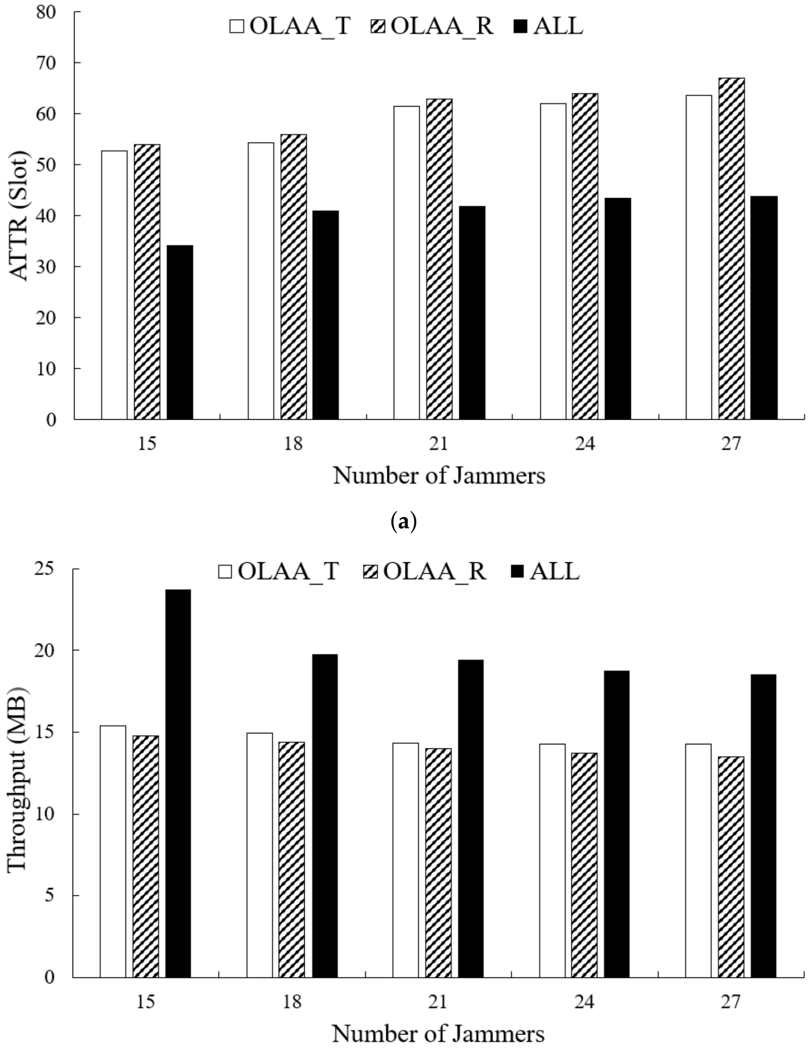


(**a**)

(**b**)

Figure 10. This figure displays the impact of the number of channels on the star topology's performance in an asynchronous network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.

(B) Impact of the number of jammers: This experiment involves altering the count of malicious attackers in an eight-channel network, as illustrated in Figure 11. The simulation outcomes demonstrate that as the number of attackers increases, the throughput reduces correspondingly, owing to a greater occurrence of jamming attacks. It is interesting to find that the throughput decreasing rate slows down when more jammers are continuously added to the network. We believe it is because when the number of attackers increases, the attacking ranges of the jammers are overlapped, which reduces the attacking efficiency. Again, being able to increase the number of rendezvous and preserve the randomness of channel allocation, the performance achieved by the ALL protocol is better than that of the OLAA_T and OLAA_R protocols.



(**a**)

(**b**)

Figure 11. This figure presents the impact of the number of jammers on the performance of mesh topology in a synchronized network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.

(C) Impact of the number of PUs: Next, we explore how the different protocols are impacted by varying numbers of PUs. As depicted in Figure 12, the performance of all three protocols remains consistent across different numbers of PUs, with ALL demonstrating noticeably superior results compared to OLAA_T and OLAA_R. The average packet delivery ratios achieved by ALL, OLAA_T, and OLAA_R under different numbers of PUs are 75%, 64%, and 65%, respectively. It is worth noting that when the number of PUs is set to four, the maximum packet delivery ratio achievable by an SU running ALL, OLAA_T, and OLAA_R is 82%, 72%, and 71%, respectively. This experiment confirms that ALL is resilient and not adversely affected by an increase in PUs.

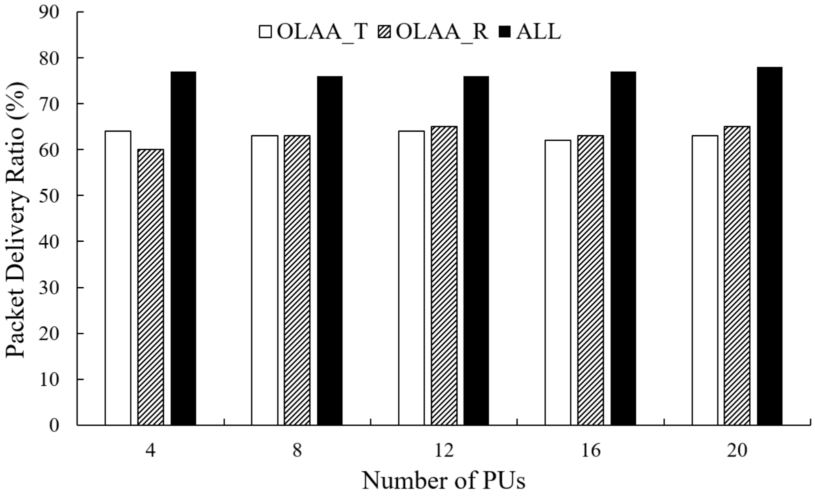
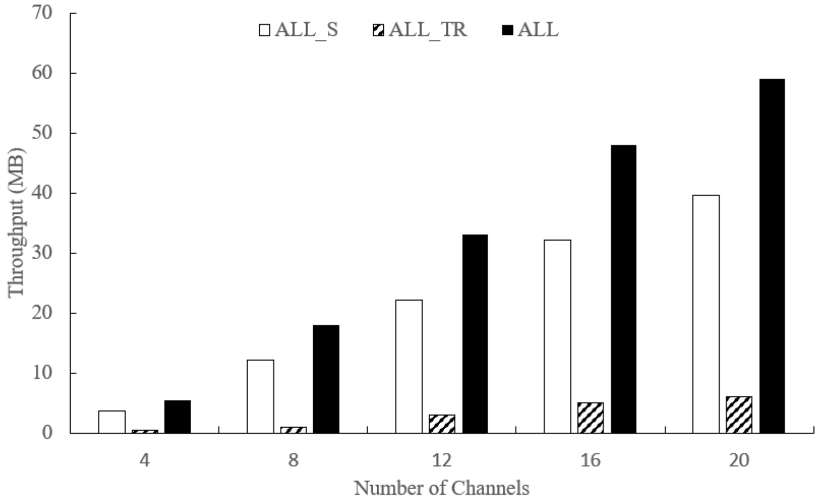


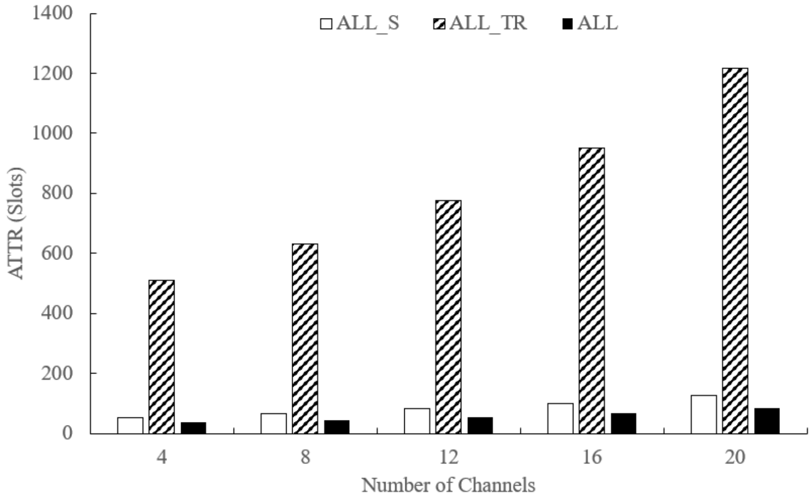
Figure 12. This figure presents the impact of the number of PUs on the performance of mesh topology in a synchronized network, as measured by the packet delivery ratio.

(D) Impact of the mechanisms of shortening ID and allocating unfixed *R* frames: The performance of the two mechanisms of the ALL protocol is being observed in this experiment: shortening the ID of SUs (the resulting protocol is denoted as ALL_S) and allocating the position of unfixed *R* frame based on receivers' hopping matrix structures (the resulting protocol is denoted as ALL_TR). The simulation results are shown in Figure 13. We can see that the performance of ALL_S and ALL_TR is lower than that of ALL. In a synchronous network, the first column of each node's channel hopping matrix overlap and thus do not provide a rendezvous guarantee. ALL_S does not perform as well as ALL because a sender does not adjust its channel hopping matrix based on those of its receivers. The performance of ALL_TR lags far behind that of ALL and ALL_S since SUs running ALL_TR have similar IDs when compared to those running ALL and ALL_S. Similar IDs produce similar column structures in channel hopping matrixes and thus reduce the probability of providing a rendezvous between SUs.



(**a**)

Figure 13. *Cont.*



(**b**)

Figure 13. This figure displays the impact of shortening ID and allocating unfixed *R* frames on the performance of mesh topology in a synchronous network, as measured by two metrics: (**a**) throughput and (**b**) ATTR.

6. Conclusions

In this paper, we propose the anti-jamming channel hopping protocol ALL, which provides a rendezvous guarantee in CRNs. The ALL protocol, which enables each SU to adjust the location of *T* and *R* frames in its channel hopping matrix based on the channel hopping structures of its receivers, is constructed using the existing OLAA_T protocol as a foundation. We have proven that the ALL protocol can ensure each SU has a rendezvous with its intended receiver on all commonly available channels. The simulation results verify that ALL can improve network performance by 25% and 20%, on average, in terms of throughput and ATTR, respectively, when compared to OLAA_T and OLAA_R.

Shortening the ID of SUs is advantageous for the ALL protocol, but it also results in multiple nodes having the same or similar shortened IDs. This could cause issues as nodes with identical shortened IDs may not be able to meet, and thus, a user cannot transmit data to another user with the same ID. This constraint may pose a challenge in a network that is not densely populated. To resolve this issue, it may be beneficial to not significantly reduce the ID length. However, determining the appropriate ID length requires further exploration. Further, we plan to further investigate the impact of the *T* frame symbol assignment strategy in the future. A possible strategy is that a sender, SU *i*, assigns *T* frame symbols such that SU *i* can have a rendezvous with most of its receivers when they are TR overlapped.

Author Contributions: Conceptualization, Y.-H.H., C.-M.C., C.-Y.L. and C.-C.Y.; methodology, Y.-H.H. and C.-M.C.; software, Y.-H.H.; validation, C.-M.C., C.-Y.L. and C.-C.Y.; formal analysis, Y.-H.H. and C.-M.C.; investigation, Y.-H.H. and C.-M.C.; resources, C.-M.C.; data curation, Y.-H.H.; writing—original draft preparation, Y.-H.H.; writing—review and editing, C.-M.C., C.-Y.L. and C.-C.Y.; visualization, Y.-H.H.; supervision, C.-M.C.; project administration, C.-M.C.; funding acquisition, C.-M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was sponsored by National Science and Technology Council, R. O. C., under grants MOST 110-2622-E-019-003 and MOST 111-2221-E-019-050.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Federal Communications Commission. *Federal Communications Commission Spectrum Policy Task Force*; Report of the Spectrum Efficiency Working Group; Federal Communications Commission: Washington, DC, USA, 2002.
2. Akyildiz, I.F.; Lee, W.Y.; Vuran, M.C.; Mohanty, S. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Comput. Netw.* **2006**, *50*, 2127–2159. [CrossRef]
3. Hou, Y.T.; Shi, Y.; Sherali, H.D. Spectrum Sharing for Multi-Hop Networking with Cognitive Radios. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 146–155. [CrossRef]
4. Cormio, C.; Chowdhury, K.R. Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping. *Ad Hoc Netw.* **2010**, *8*, 430–438. [CrossRef]
5. Kim, M.R.; Yoo, S.J. Distributed coordination protocol for ad hoc cognitive radio networks. *J. Commun. Netw.* **2012**, *14*, 51–62. [CrossRef]
6. Lo, B.F.; Akyildiz, I.F.; Al-Dhelaan, A.M. Efficient Recovery Control Channel Design in Cognitive Radio Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2010**, *59*, 4513–4526. [CrossRef]
7. Wang, X.; Zhang, X.; Zhang, Q.; Tang, C. Common control channel model on MAC protocols in cognitive radio networks. In Proceedings of the 2011 International Conference on Computer Science and Network Technology, Harbin, China, 24–26 December 2011; pp. 2230–2234.
8. Chao, C.M.; Fu, H.Y.; Zhang, L.R. A Fast Rendezvous-Guarantee Channel Hopping Protocol for Cognitive Radio Networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 5804–5816. [CrossRef]
9. Chao, C.M.; Fu, H.Y. Supporting Fast Rendezvous Guarantee by Randomized Quorum and Latin Square for Cognitive Radio Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 8388–8399. [CrossRef]
10. Chao, C.M.; Fu, H.Y. A fast and fair rendezvous guarantee channel hopping protocol for cognitive radio networks. In Proceedings of the 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), Beijing, China, 4–6 June 2016; pp. 483–487.
11. Wu, S.H.; Wu, C.C.; Hon, W.K.; Shin, K.G. Rendezvous for heterogeneous spectrum-agile devices. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 2247–2255.
12. Wu, C.C.; Wu, S.H. On Bridging the Gap between Homogeneous and Heterogeneous Rendezvous Schemes for Cognitive Radios. In Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Bangalore, India, 29 July–1 August 2013; pp. 207–216.
13. Chen, L.; Bian, K.; Chen, L.; Liu, C.; Park, J.M.J.; Li, X. A Group-Theoretic Framework for Rendezvous in Heterogeneous Cognitive Radio Networks. In Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Philadelphia, PA, USA, 11–14 August 2014; pp. 165–174.
14. Gu, Z.; Pu, H.; Hua, Q.S.; Lau, F.C.M. Improved rendezvous algorithms for heterogeneous cognitive radio networks. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, 26 April–1 May 2015; pp. 154–162.
15. Liu, H.; Lin, Z.; Chu, X.; Leung, Y.W. Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1867–1881. [CrossRef]
16. Paul, R.; Choi, Y.J. Adaptive Rendezvous for Heterogeneous Channel Environments in Cognitive Radio Networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 7753–7765. [CrossRef]
17. Chao, C.M.; Chen, C.T. Ratio adjustable channel hopping enhancement for heterogeneous cognitive radio networks. In Proceedings of the 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, 21–25 May 2017; pp. 1093–1098.
18. Chao, C.M.; Chen, C.T.; Huang, S.J. An Adjustable Channel Hopping Algorithm Based on Channel Usage Ratio for Multi-Radio CRNs. In Proceedings of the 2019 International Conference on Information Networking (ICOIN), Kuala Lumpur, Malaysia, 9–11 January 2019; pp. 307–309.
19. Chao, C.M.; Fu, H.Y. Supporting fast and fair rendezvous for cognitive radio networks. *J. Netw. Comput. Appl.* **2018**, *113*, 98–108. [CrossRef]
20. Chao, C.M.; Lee, W.C. Load-aware anti-jamming channel hopping design for cognitive radio networks. *Comput. Netw.* **2021**, *184*, 107681. [CrossRef]
21. Chen, K.W.; Chao, C.M.; Lin, C.Y.; Yeh, C.C. Anti-jamming channel hopping protocol design based on channel occupancy probability for Cognitive Radio Networks. *Comput. Netw.* **2022**, *214*, 109125. [CrossRef]
22. Wang, Q.; Xu, P.; Ren, K.; Li, X.-y. Delay-bounded adaptive UFH-based anti-jamming wireless communication. In Proceedings of the 2011 Proceedings IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 1413–1421.
23. Lee, E.K.; Oh, S.Y.; Gerla, M. Frequency Quorum Rendezvous for Fast and Resilient Key Establishment under Jamming Attack. *SIGMOBILE Mob. Comput. Commun. Rev.* **2010**, *14*, 1–3. [CrossRef]
24. Oh, Y.H.; Thuente, D.J. Limitations of Quorum-based Rendezvous and key establishment schemes against sophisticated jamming attacks. In Proceedings of the MILCOM 2012—2012 IEEE Military Communications Conference, Orlando, FL, USA, 29 October–1 November 2012; pp. 1–6.
25. Poisel, R. *Modern Communications Jamming Principles and Techniques*, 2nd ed.; Artech: New York, NY, USA, 2011.
26. Viterbi, A.J. *CDMA: Principles of Spread Spectrum Communication*; Pearson: New York, NY, USA, 1995.

27. Strasser, M.; Popper, C.; Capkun, S.; Cagalj, M. Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, 18–21 May 2008; pp. 64–78.
28. Wu, Y.; Wang, B.; Liu, K.J.R.; Clancy, T.C. Anti-Jamming Games in Multi-Channel Cognitive Radio Networks. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 4–15. [CrossRef]
29. Huang, J.F.; Chang, G.Y.; Huang, J.X. Anti-Jamming Rendezvous Scheme for Cognitive Radio Networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 648–661. [CrossRef]
30. Chang, G.Y.; Wang, S.Y.; Liu, Y.X. A Jamming-Resistant Channel Hopping Scheme for Cognitive Radio Networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 6712–6725. [CrossRef]
31. Sharma, H.; Kumar, N. Deep learning based physical layer security for terrestrial communications in 5G and beyond networks: A survey. *Phys. Commun.* **2023**, *57*, 102002. [CrossRef]
32. Naderi, E.; Asrari, A. Experimental Validation of a Remedial Action via Hardware-in-the-loop System Against Cyberattacks Targeting a Lab-scale PV/Wind Microgrid. *IEEE Trans. Smart Grid* **2023**, 1.
33. Linek, V.; Jiang, Z. Extended Langford Sequences with Small Defects. *J. Comb. Theory Ser. A* **1998**, *84*, 38–54. [CrossRef]
34. Langford, C.D. Problem. *Math. Gaz.* **1958**, *42*, 228. [CrossRef]
35. Simpson, J.E. Langford sequences: Perfect and hooked. *Discret. Math.* **1983**, *44*, 97–104. [CrossRef]
36. Moulahi, T.; Nasri, S.; Guyennet, H. Amelioration of MPR by a backbone-based broadcasting algorithm for WSNs. In Proceedings of the 2012 International Conference on Information Technology and e-Services, Sousse, Tunisia, 24–26 March 2012; pp. 1–5.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.